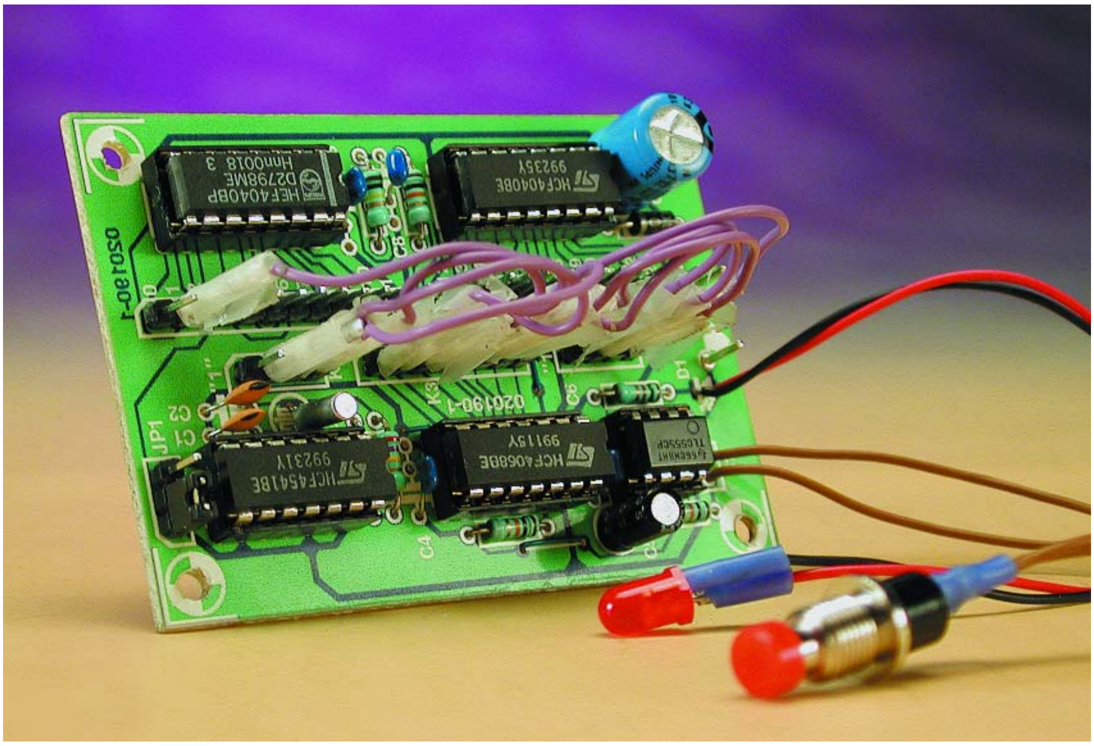
Week/Month Timer

Programmable for long periods

From an idea by Doug Pratt

As opposed to most timer circuits, this month's Mini Project is intended for very long periods. A switchable divider chain using two 4040 counter ICs allows periods of up to 194 days (yes that's 6 months) to be timed with great accuracy. Almost any other period shorter than the maximum may be set up, and all without a microcontroller or indeed any sort of exotic component!



It started out as a relatively simple project — a timer to remind us (and friends) to take our wheelie bins out ready for collection once a week (or month). The original intention was to make the timer drive a voice/record playback module programmed with a snappy message. However, for the purpose of this Mini Project, such an extension was deemed too complicated, so a rather simpler 'time-out' warning device was chosen, although it is still possible to hook up a voice module if you want to.

The vast majority of circuits capable of counting down predefined periods are intended for use as a cooking timer, darkroom timer or parking timer. Their timing range will typically cover a few minutes to a couple of hours. If you want to be reminded of your fortnightly wheelie bin chore, the 6-weekly appointment at the hairdressers, the 3-monthly check-up at the GP, or an equipment servicing interval, then you need to resort to a calendar or a diary because timers with such long periods simply do not exist. Oops, that should be "did not exist" because the timer described in this article is the exception to the rule. Here's why: it can be set to periods **within the range 2 to 16,777,216 seconds!**

Simple electronics

The circuit diagram (schematic) of the timer is given in **Figure 1**. As you can see, the project consists of just five plain integrated circuits and

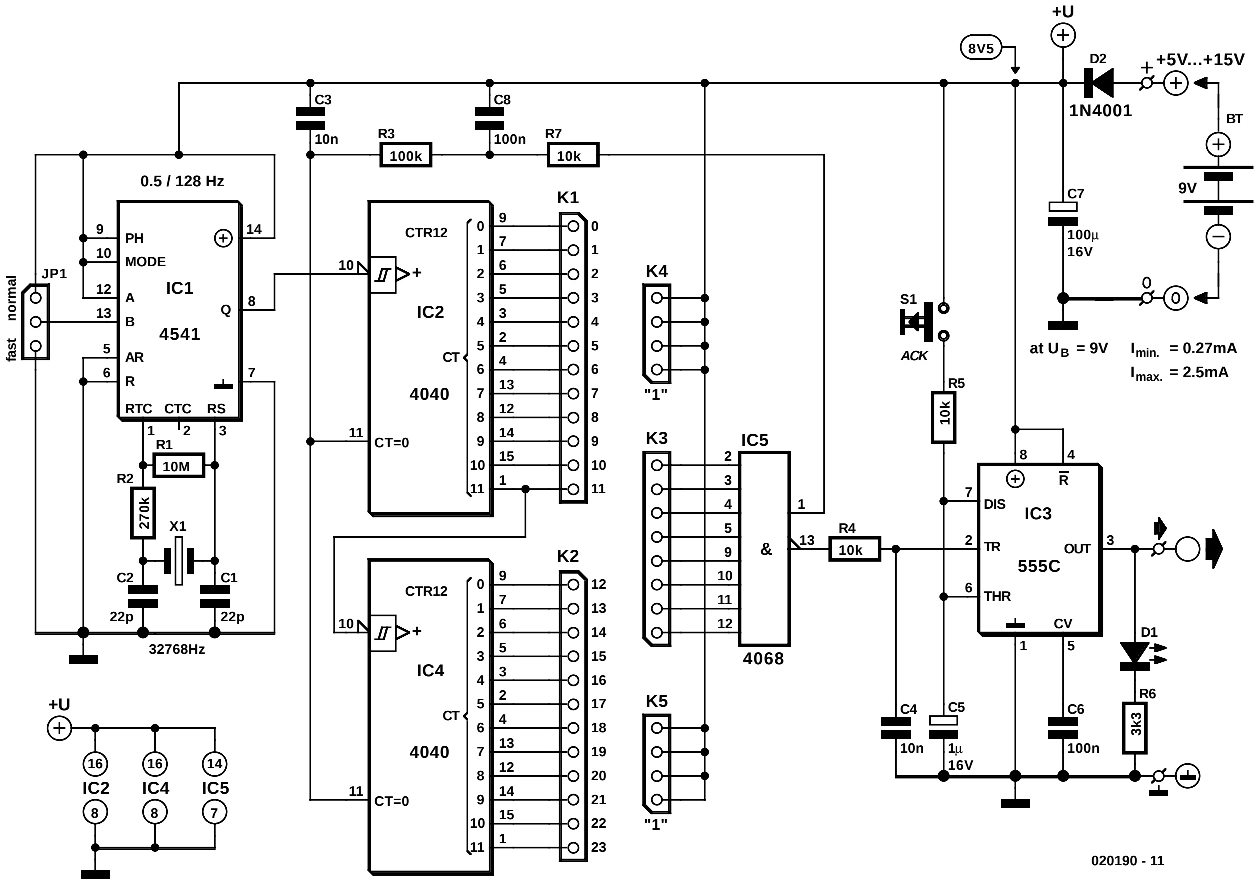


Figure 1. Cascaded binary counters enable the circuit to generate extremely long delays like weeks and months with 1-second accuracy.

a handful of discrete parts. Functionally, the circuit may be divided into three sections: an oscillator, a user-configurable counter and a flip-flop in control of an LED which acts as an indicator that the programmed period has elapsed. Note that the circuit allows for other visual or acoustic actuators to be controlled, including a buzzer.

IC1 and its surrounding passive components acts as the oscillator. The 4541 is a little known CMOS IC described by the manufacturers as a 'programmable timer'. In addition to an oscillator it contains a 16-stage binary counter which may be preset using pins A and B on the chip. Here, a divisor of 2^{16} has been chosen. Consequently, the quartz-generated clock frequency of 32,768 Hz is divided down to 0.5 Hz, which corresponds to a period of 2 seconds.

The 2-seconds pulse in turn clocks a divider chain ('cascade') consisting of two 12-stage binary counters type 4040 (IC2 and IC4). The ICs are cascaded by connecting the last output (Q11 on IC2) of the first IC to the clock input (CLK) of the second counter (IC4). In this configuration, the two 4040s allow a divisor of 2^{24} to be set, which results in the previously mentioned maximum time of 16,777,216 seconds. At this point some of you may wonder why the period is not twice as long — after all, the divider chain is fed with 0.5 Hz and not 1 Hz. The answer is simple: the output is activated as soon as **half** the period time of the counter has elapsed.

The divider cascade is followed by an 8-input NAND gate (IC5). By connecting one or more counter outputs (available on K1 and K2) to the input(s) of the NAND gate (available on K3), almost any desired period can be defined.

Once the preset value ('count') is reached, two things happen: to begin with, the Low-to-High transition that occurs at the non-inverting output of NAND IC5 is used to reset our two counters IC2 and IC4 by way of two R-C networks R7-C8 and

Table 1. Divisors.

Connection	seconds
K1-0 (Q0-IC2)	2
K1-1 (Q1-IC2)	4
K1-2 (Q2-IC2)	8
K1-3 (Q3-IC2)	16
K1-4 (Q4-IC2)	32
K1-5 (Q5-IC2)	64
K1-6 (Q6-IC2)	128
K1-7 (Q7-IC2)	256
K1-8 (Q8-IC2)	512
K1-9 (Q9-IC2)	1,024
K1-10 (Q10-IC2)	2,048
K1-11 (Q11-IC2)	4,096
K2-12 (Q0-IC4)	8,192
K2-13 (Q1-IC4)	16,384
K2-14 (Q2-IC4)	32,768
K2-15 (Q3-IC4)	65,536
K2-16 (Q4-IC4)	131,072
K2-17 (Q5-IC4)	262,144
K2-18 (Q6-IC4)	524,288
K2-19 (Q7-IC4)	1,048,576
K2-20 (Q8-IC4)	2,097,152
K2-21 (Q9-IC4)	4,194,304
K2-22 (Q10-IC4)	8,388,608
K2-23 (Q11-IC4)	16,777,216

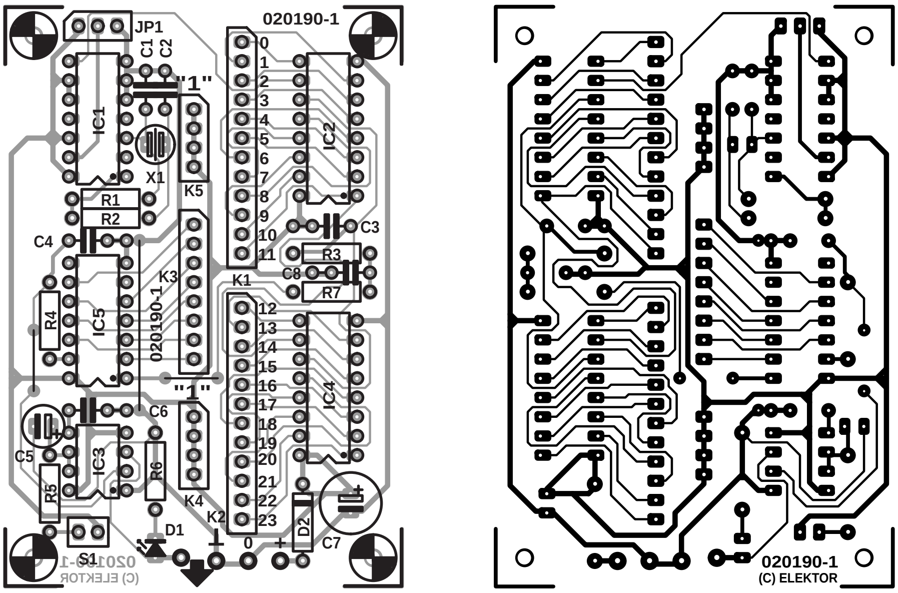


Figure 2. The PCB designed for the Week/Month Timer is small and together with the 9-V PP3 battery will easily fit in a standard plastic case.

COMPONENTS LIST

Resistors:
R1 = 10MΩ
R2 = 270kΩ
R3 = 100kΩ
R4,R5,R7 = 10kΩ
R6 = 3kΩ3

Capacitors:
C1,C2 = 22pF
C3,C4 = 10nF
C5 = 1µF 16V radial
C6,C8 = 100nF
C7 = 100µF 16V radial

Semiconductors:
D1 = LED, high-efficiency, 2 mA
D2 = 1N4001
IC1 = 4541
IC2,IC4 = 4040
IC5 = 4068
IC6 = 555C (CMOS version)

Miscellaneous:
JP1 = jumper
K1,K2 = 12-way SIL pinheader
K3 = 8-way SIL pinheader
K4,K5 = 4-way SIL pinheader
S1 = pushbutton, 1 make contact
X1 = 32.768kHz quartz crystal
9-V PP3 (6R22) battery with clip-on wires
PCB, available from The PCBShop (see Elektor website)

R3-C3. This causes the user-defined timing period to be started over again. Secondly, the High-to-Low transition at the inverting output of IC5 is fed to the flip-flop in order to trigger it.

IC5, a 555 timer in so-called bistable mode acts as a flip-flop. The flip-flop will toggle, pulling its output High, on the falling edge of the pulse applied to the TR– input. Its pevious state is restored when a positive pulse transition (rising edge) at the TR+ input. So, when the preset period has elapsed on the timer, the output of IC3 goes High causing LED D1 to light. If you are not satisfied with just the LED, the same output may be used to drive an active buzzer or another low-power actuator. The user can acknowledge the 'time-out' warning (whether visual or acoustic) by pressing pushbutton S1. This causes a positive pulse to be applied to the TR– input and the flip-flop to return to its non-active state.

Some more details

A couple of details in the circuit have escaped discussion so far.

As indicated in the circuit diagram, jumper JP1 provides a selection between 'fast' and 'normal'. This extra feature has been added for test purposes. When 'fast' is selected, pin B of IC1 is pulled to ground. This causes the divisor in IC1 to be changed such that the output frequency at the Q pin is increased by a factor of 128. This is useful when testing the circuit because you don't have to wait so long to see you got the 'programming' right (more about this further on).

Because the counters work a-synchronously (they all operate in 'ripple' mode i.e., consecutively), glitches may occur at the output of NAND gate IC5 and these may cause false timing results. The problem is solved by R4-C4 which suppress glitches and spikes. The propagation delay of each counter in the 4040 ICs is slightly dependent on the actual supply voltage and will vary between 50 ns and 100 ns per counter stage. For IC2 and IC4 together, this works out at a maximum of (about) 24 × 100 ns = 2.4 µs.

Of course, the non-inverting output of the NAND gate may also produce glitches which when not suppressed could easily cause false resetting of the counters. Here, too, an R-C network is in order, this however has a potential problem because it causes the reset voltage to increase slowly. Assuming that the two counters are not exactly identical (which unfortunately is always the case) then one of them will be reset while the reset threshold for the other is not yet reached. That's annoying because the resetting of one of the counters will cause the reset pulse to disappear, resulting in just one of the counters being cleared to zero and the other still at its previous count. To prevent this from happening, a double R-C network R7-C8, R3-C3 is applied which gives the reset signal a quite different shape: there now is a 'dead' time, which is another way of saying that the entire pulse is slightly delayed. The upshot is that it not only takes a little longer for the pulse to reach the reset level, but also that the pulse is allowed to 'rise' further by about 0.5 V, to make absolutely sure the other counter is also reset.

How to set the period

Calculating which outputs you have to use to set up a certain period is quite simple, the process being similar to doing a long division.

Step 1. Calculate the desired time in seconds. Let's take one week as an example: 60 seconds × 60 minutes × 24 hours × 7 days = 604,800 seconds.

Step 2. Use the divisor table (**Table**

1) to look up the first number that's smaller than the above value. You'll find Q4 on IC4 with the associated divisor 524,288. Now determine the difference:
604,800 – 524,288 = 80,512

Step 3. Repeat step 2 until the remainder is zero:
80,512 – 65,536 = 14,976
(output Q1 on IC4)
14,976 – 8,192 = 6,784
(output Q0 on IC4)
6,784 – 4,096 = 2,688
(output Q11 on IC2)
2,688 – 2,048 = 640
(output Q10 on IC2)
640 – 512 = 128 (output Q8 on IC2)
128 – 128 = 0 (output Q6 on IC2)

Each of the counter outputs found in this way has to be connected to a NAND gate input. Any non-used inputs on the gate have to be tied to the positive supply rail, hence the presence of connector strips K4 and K5.

Construction

The artwork designed for the printed circuit board is shown in **Figure 2**. The ready-made board is not available from Readers Services but may still be obtained through an alternative channel called **The PCBShop** (EuroCircuits). Alternatively you may want to etch your own board using the artwork printed here (or downloaded as a .pdf file from our website). The construction itself should not cause problems even to relative newcomers.

If you intend to use the counter for just one, fixed, period, then the necessary links between K1/K2 and K3 may be simple wires permanently installed on the circuit board. The same goes for the unused inputs on the NAND gate.

On the other hand, if you (like us) envisage having to use different long periods, it is better to fit SIL pin-headers in positions K1-K5. Selecting the desired period is then easily done using a set of short cables with a pin connector at each side. The pin connectors should be isolated using heat shrink sleeving or similar.

Power supply

The Week/Month Timer described here is very economical in use, drawing just 0.3 mA with the LED off. When the LED comes on, the current drain rises to about 2 mA. Obviously, low power consumption is a must because in practice the circuit will be powered from a battery. Today's PP3 9-V alkaline batteries have a nominal capacity of about 500 mAh which is sufficient for about two months of continuous operation of the timer (provided of course you do not leave the LED on for weeks on end). If this sort of battery capacity is insufficient for your application, consider using a pack of four penlight (AA) batteries in a holder, or a mains adapter with a stabilized output voltage between 5 and 15 volts dc.

(020190-1)